# Coursework Guideline

### Documentation Headers
### (please copy them and paste into your own documentation file)

### Problem
Problem identification
Objectives to achieve

### Design
Planning of work
Description of Hardware/Software

### Output
Sample Runs / Hardcopy / Report(s) / List

### Technical Documentation
Explanation of programming / scripts / buttons
Use of functions / procedures
Description of data (Database fields,  etc.)
System / Program-description
User manual /help facilities

### Coursework Evaluation
Self evaluation
Options for improvement

# How to make the documentation about your coursework (school-based internal assessment for IGCSE - computer studies, Cambridge)

As you all know, part of the course “Computer Studies” is the practical coursework. The official name is the “internal assessment”. Internal means, that the School and teacher involved will do the marking of this coursework! The school will have to send (10 samples of) the coursework to Cambridge, where they will be rechecked (moderated) and marked again. If they think the marking done was not adequate, they will adjust the (all!) grades! Therefore it is extremely important to follow these coursework guidelines extremely well! We will use this guideline for all in between and final grading! We strongly advise you to copy the headers from the first page into your documentation. This is the best way not to forget an important part. Realise as well, that the coursework (only!) counts for **25** % and the theory exam (with a $2^{1/2}$ hours paper!) for **75** % of your final IGCSE exam-grade.

Unfortunately we are not allowed to send floppy disks with your program on it to Cambridge. Your coursework must be handed in / documentated on ***PAPER only!*** This means, that you will have to spend a lot of your time doing wordprocessing, describing and explaining what you have been doing and what the results are. That also means that a good documentation is as important as a good program! In fact, you will maybe even spend more time doing the documentation as you worked on the program (e.g. database or website) itself! So therefore, be sure to know how the things in your computer lab work. You should be familiar with (in doubt? Ask the teacher or another student):

- Using a screen capture facility (on the Mac: Apple/Shift/3)
- Using the spell check / thesaurus / webster’s (all compulsory!)
- Making headers / footers (please include your name and (!) date!)
- Saving your work on a regular base, making backup’s
- Using the scanner (minimising the size of files etc.!)
- Using the Internet (how to download / copy text and pictures)
- Finding old examples of coursework (see white folders in class)
- Places to find more info (IGCSE folder on the intranet Server)
- The resources in the Library (at school and outside school!)

- The IGCSE website on the Internet
  go.to/igcse
- The IGCSE Yahoo club on the net
  http://clubs.yahoo.com/clubs/igcsecomputerstudies
- You should know the deadlines announced (see Time schedule)
- And last but least, always use this guideline!

**(Some) Software you could use for your coursework:**

- Calculations for business / commerce / finance / grades / etc. with the help of a SPREADSHEET (e.g. Microsoft Excel)
- Storage and retrieval of information in a DATABASE (e.g. Claris FileMaker Pro, Microsoft Access)
- Multimedia programming (animation, storage of data) with HYPERCARD and its programming language HYPERTALK
  (Hyperstudio / Metacard for windows / mac are great as well)
- Professional software (also called Authoring Tools) for creating multimedia software, e.g. for educational uses. AuthorWare or Macromind Media Director are some good examples.
- Desktop publishing with sophisticated software like Adobe Illustrator, Aldus Freehand, Adobe Photoshop, Quark XPress
- ClarisWorks, Microsoft Word or WordPerfect for the documentation of your coursework
- Web-page design (WWW-Internet). Using html-language and layout programs (Claris Homepage, Adobe Pagemill, Microsoft Frontpage). Animations and Java-scripting (programming)
- Making an application with programming languages (C++, Visual Basic, Pascal or others).
- Creating a sophisticated multimedia Presentation

Check out before you make your decision if the software is available in the Lab. Maybe the teacher needs to install some features that are not standard in the lab. Also think about the software you can use at home. If the systems at school and at home are not the same (e.g. Apple Mac’s and Windows) consider software that runs well on both platforms: Microsoft office products (Excel, Powerpoint, Word), Claris products (Clarisworks, Filemaker Pro database and Claris Homepage). Metacard (http://www.metacard.com) is the (much improved!) alike of Hypercard (object orientated programming) and is also supporting multiple platforms. Check with your teacher how you can exchange files between Home and School, at the ISH we support 1.4 MB and 120 MB Floppies, 100 MB Zip-drives, E-mail attachments and CD-Rom. If you write your own CD-RW disk, make sure you finish (close) the session, otherwise you can only see the contents of the disk on your own computer. . .

Install at home a screen-capture utility as well, so you can make screen-copies of work done at home as well.

You can do major parts of the work at home, but make sure that you bring the work to school on a regular base. Some students prefer to do the documentation at home, some prefer to work more at school.

Ask friends and other students you know, what they did for their coursework and how much time was involved! (Most of them will tell you they totally underestimated the time needed . . .)

Normal lessons times are 2 hours theory and one hour practical a week. The total amount of practical hours in year one would then be: 40 * 1 = 40 hours. Now subtract some missing periods (exams, free’s for whatsoever reason etc.) and you see, it ain’t much, about 32 hours in total. Just a few days in a row, ever wondered how fast a week holiday is over?? Well it’s the same with the time for practical . . . Your second year is even shorter (some 23 practical hours only!), your Mock exams will be starting in January, last deadline for coursework is usually around the very beginning of April!

## Necessary Contents Of Your Coursework

For each logical part of your coursework documentation you can get a maximum of points. The total score of the coursework can be 50 points. In the following explanation you can see how much you can get (maximum / minimum) in detail. All the important parts you should cover are listed. Please read the instructions, use this guideline as what it is: a guideline!

ATTENTION: the points for this marking scheme are different compared to the official scheme provided by Cambridge. We award less points for programming, more points for Design and Layout

## PROBLEM DESCRIPTION

### Problem Identification

☞ Description of the problem you will solve with your coursework, **disadvantages** of the old system (how was it done before your computer solution took over . . . .)

- Explain how you got the idea for your coursework
- Who has the problem and why
- Describe any other alternative solutions for the problem possible, what would be the (dis)advantages of other solutions? You can describe other manual solutions, but you can also refer to other computer-based solutions. Think about aspects as costs, how complicated it would be, how fast a solution could be implemented, what training is required . . . .
- Describe what's wrong with the present solution / situation, how would you (do they) solve it (so far) without using a computer (system)

Please make sure to get the full 2 points here. Compare before and after (with or without the computer solution). Try to list as many disadvantages of the situation before you came up with a computerised solution to the problem. If you create a program just for yourself, use your imagination, pretend maybe that you are making your program for a company! You may exaggerate here a little, it doesn't have to be for real really . .

☞ Describe how the situation will change with the help of your computer solution. What are the **advantages** of using the new (your) system?

- Explain how far the new system will make life "easier". Mention the things that your program can (will) do for better, try to list also the advantages for the daily business, it is o.k. to make a link here to business / eco! Will the system be of an advantage for customers too? Think about what your output can do (reports, overviews, bills, mailmerge letters, charts with percentages, sales / costs / profit analysis, different print sizes (labels / business cards), labour costs, think about aspects as marketing (worldwide on the Internet maybe?), professional presentation, competition, social impacts . . . . . . Just don't be too shy here. Imagine your program would be very successful and people would be using it for real (and you will end up as a rich person . . )

☞ You should really investigate your problem. Go and ask the people involved (library, shop, canteen, teachers etc.!). Make use of questionnaires if appropriate, go and visit a shop (where they rent out CD's for example) etc. You will see, most people are really happy to show you around, give you some papers or other materials, just do it!

0/1/2 points

**Objectives to achieve**

What are your goals, your aims, what do you want to solve with the computer program? What do you propose? Explain what you expect to do! Try to relate this to your problem description, try to be realistic, but also state specifically what you will create (don't tell here e.g. what fields you will use in your database, that comes later, but list here what functions you will offer (like a detailed report, a menu, a place to see books not returned yet (library), a list of all customers, a mailmerge for writing letters (bills?), a search facility, a template for sticky labels, password protection, a different layout (mode / interface) for shop-users and shop-owners etc. etc. Have a look at examples, hear around, surf the web, go to shops, ask your parents (yeah, that could help too . .!), maybe somebody will give you the ultimate idea for a great coursework, that will be off practical use! There are a dozen of shops in your neighbourhood, that still don't have a web site yet, or don't even know what a database is. Don't underestimate yourself!

0/1/2 points

**DESIGN**

**Planning of the coursework**

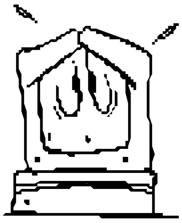
☚ Describe your way of setting up your coursework. What was your overall plan? Explain in logical and / or chronological order!

☚ How did you analyse and investigate the problem (Whom did you ask for information and how, did you have inquiries, interviews with people, questionnaires, . . ?)

☚ How did you collect your data for INPUT necessary for your solution (what data / information is needed / used in your coursework?)

☚ How did you investigate how the OUTPUT of your program should be? Did you compare to existing other solutions maybe? Remember, a little fantasy can do no harm . . .

☚ How did you plan your actual programming / use of the package? Did you make some (simple) flowcharts? (They always give extra credit!!).



☞ What other steps did you make in the actual implementation (programming) of your coursework? How did you decide about which software to use? Did you compare some alternatives maybe?

☞ Mention here steps you did, like: testing, evaluating, improving your program?

☞ What difficulties and problems did you encounter? What steps did you take to overcome them?

In general, try to make here a nice list of planning steps you (could have . . ) did. It should look professional, try to follow the program life-cycle (remember, you should know about system design / analysis, feasibility etc.!) Give it all a nice layout, you could use a flowchart program to show your planning stages (Powerpoint does a good job there!). A good overview is really helpful, distinguish between planning and realisation.

0/1/2/3 points

**Description of Hardware and Software used**

☞ Describe the different stages in your work. Explain (for each stage) **WHAT** SOFTWARE you used, **WHEN, HOW** and **WHY** (e.g. a word-processor to describe your problem, to make some enquiry-questions, a spreadsheet to produce CHARTS, a GRAPHICAL application to draw flowcharts, a SCREEN GRABBER to capture images, Photoshop to change images, the Internet to get information, E-mail . . .

☞ What HARDWARE did you use, when, how and why, what hardware would be necessary for your computer solution, what hardware could be used in addition. Describe the used system and peripherals, all input and output devices. Think about the hardware that you could (!) use with your coursework (maybe at a later stage, when money is not a problem). Think about scanners, Internet connections, digital cameras, touch screens, local and wide area networks (modem, ISDN). (Just include your last years birthday wish list you gave your parents . . . )

Think about backup systems, like CD-ROM writers, removable disks, tape streamers, optical storage. And last but not least, mention things that could be used (later maybe?!) for better data input, like: OCR, OMR, bar code reading / scanning. Can voice recognition be of use for your system? Automatic translation maybe? (check Altavista on the Internet to see practical examples of that technology!) etc. etc.

Just a simple list (I used a keyboard, a computer and a mouse . . .) will not be awarded any points!! Some detailed info about our lab can be found on the intranet.

0/1/2/3 points

**OUTPUT (of Reports, Lists, Charts, sample Runs etc.)**

☞ Add in your documentation your SAMPLE RUNS (hardcopies that show the actual working of your program)

Points will be given for the availability of sample runs of your program. Here you have to prove that your solution works and you need to show HOW it works! The sample runs should also be part of your user-documentation. On the MAC’s you can make a screen-dump by pressing the <shift>-key, the <>-key and 3 (all at the same time!) With the mouse you then can select any part of the screen you want. It will be saved in the scrapbook (or clipboard) and from there it can be easily imported in your documentation (copy and paste). Don’t forget to empty the scrapbook after use please!

If you use a database program, make sure you present hardcopies of all records in a columnar report and some records in a full screen report.

☞ Print out and explain with hardcopies your solution, showing how you designed and implemented your computer solution.

That means: don’t just print out a report (or database list), but also explain in words what the purpose of the list is (can be). Remember, the Cambridge people will only see your documentation, they won’t see your program, unless you provide that printed out on paper! And don’t expect them to guess what it is all about. Treat them like dummies, explain all bits and pieces!

☞ Show and (!) explain your LISTS and / or REPORTS of output data (especially when you use a Database or Spreadsheet). All software allows data output. You should print out examples of everything you achieved. Don’t leave anything out, as simple as you even think it might be! You should (of course!) provide also lists of all fields (cells, variables . . ) all formulas (scripts, functions . . ) and all other things you used / achieved. Remember, 5 points is a lot, you need to prepare and show something relevant here as well!

For example in a database like FileMaker you should show that you can use the different lay-outs for reports. Show at least ONE COLUMNAR report (with several fields in a row and the data in columns). Don’t forget to SORT your data before you PRINT! Show that you can use different FIND’s (query) for your data and print out the query itself and the records found. Spreadsheets should make use of a nice design (show that you know how to print WITHOUT grid-lines and make use of BORDERS, TEXT BOXES etc. Your charts should look well both in colour as Gray-scale. Also here use HEADERS and FOOTERS! Use page layout in Spreadsheets, data should fit on whole pages! Protect cells, show all formulas etc. etc.

0/1/2/3/4/5 points

## PRESENTATION

☞ Quality of presentation

Did you create a good lay-out, is it easy to read, well presented, nice designed in total? Don’t forget a cover page (give a title to your course work) and add a contents page. You should make good use of the wordprocessing facilities that are available nowadays. Important, don’t use more than one space at a time! Use the tab-key (or ruler) instead. Make tables, use 2 or 3 different fonts only (headers and text e.g.) Check out how to define and use different styles! Don’t forget to make use of a header and footer. It helps the teacher a lot, if you put your name and (!) date in the footer of each printed document. This allows teachers to see who’s work is being checking and which version is looked at)

Don’t even think about printing without doing a spellcheck before! Make sure that your work is clear to the reader. Provide empty pages, just stating what the next pages are about! Use empty lines to separate paragraphs. Have a look (do so!) at some very good examples to get ideas!

0/1/2/3/4 points

☞ Use of Information Technology Tools (IT-tools)

Did you use tools and software for: Spelling check, grammar check, making flowcharts / system charts / graphics, screen capture, Presentation, Internet, E-mail, Data conversion etc.? We will evaluate this, you can be smart and provide in your documentation a place where you list all the tools and utilities you have used during the making of your coursework. Don’t forget to list all the beautiful things you did at home (on dad’s super IBM (I buy Macintosh) computer ?)

0/1/2/3 points

## TECHNICAL EVALUATION

☞ Technical skills

This is the place where you will be awarded points for your technical skills. Here we evaluate as to what level the candidate is using the possibilities of the used software.

> You will get points here for the way you managed to do your coursework. Did you use only simple or also some more complicated facilities in the software? Think of things like using SORT-routines, UPDATE of records, using MAILMERGE, giving the user of your program MENUS, using CHARTS to show your results in a good-looking way, using HYPERTALK for better use of HyperCard (anybody can just go from card to card . . , using POP-UP-MENUS, VALUE-LISTS etc. in FileMaker Pro, did you use HEADER, FOOTER, FOOTNOTES, PAGE NUMBER, INDEX etc. in your wordprocessed documentation? Did you use buttons and scripts in FileMaker? Did you add some Java-scripting to your web-page?
>
> 0/1/2/3/4 points

☞ Quality of the solution

Here it is checked if your solution is appropriate and plausible. You will get points awarded reflecting how far your solution is adequate to your problem description. Remember, people in Cambridge won’t see your program, they have to deal with paper printouts only. So here you will be marked for your overall solution!

> We will check if you actually did the work on the computer, if it is of the appropriate level etc. Did you do a nice piece of work? Is it all adequate, or all very simple? Does it meet your capabilities?
>
> 0/1/2/3 points

☞ Clarity of solution

Here is checked if the solution is made clear by the candidate. Is the documentation showing what the candidate achieved? Is the solution a good one, can any novice and not-experienced user tackle the program (or database, spreadsheet). Does a user know what to do when he / she starts the program? Is there help provided? Is there any good logic in the solution of the student?

> We will check here, if your solution makes good sense. Here (important!!) it is evaluated if your final coursework meets your goal / aims (you mentioned long way before). Also we will check if your documentation is clear and describes what you have done in an understandable and logical / chronological way.
>
> 0/1/2/3 points

☞ User friendliness of the result

Is your solution user friendly? Can it easily be used by a new and not experienced user? Did you make use of value lists, Pop-up fields, menu buttons, styled layout, not too many fields / cells etc. on one page etc. etc.

> We will check here if you have thought about the later use of your

program. Will the user understand what to do? Is there any user help on the screen available? Is it all obvious to someone who needs to work on his own with your program?

0/1/2 points

## TECHNICAL DOCUMENTATION

☞ Explanation of all programming and scripting

Are you explaining all your programming, scripting and other programming techniques? You should explain:

- The working of all your scripts (e.g. in HyperCard, Filemaker, MS Access)
- The functions and formulas used in your Spreadsheet (what, why)
- The functions, scripting, calculations, buttons etc. used in your Database or other software program.

You can get 4 points here, remember (again!) that all you did, will only be known to the moderators in Cambridge if you describe / explain it in your documentation! Please really list all you have done, even if you think it is trivial. This is the spot where you can show (and prove) your computer skills!

0/1/2/3/4 points

☞ Description of Data

Did you describe all your used Data in your coursework? Think about FILES, FIELDS, VARIABLES, DATA FLOW. Explain why you have used certain data. List if data used is of a certain type / kind (e.g. numerical, text, date, formula, value list, has a range (or other validation rules) etc.! Why did you use each data? Explain it! (Filemaker allows to print a list of all data fields, print and include it as attachment!

0/1/2 points

☞ System / Program-description

Here you should explain the overall working of your program / solution. Describe any used modules, routines, menu-structures, algorithms. Make a flowchart that shows an overview of all components of your coursework (Files / Webpages etc.)

Show how you did your data testing. Prove that your program works both with right and wrong input data. Show your validation-rules and show prove (hard-copies) of how it works.

Testing a program is always necessary. For a database e.g. you will have to set-up validation rules (e.g. AGE cannot be negative, cannot be more than 120, TEST RESULT should be in between 0 and 100 etc. etc.). If you have a spreadsheet, database (anything else as well), then show what happens if you enter extreme (very large, very small or very wrong) data.

0/1/2 points

☚ User Manual / Help facilities

Here you show the "User Manual" for an unexperienced user of your program. In general that should be a few pages with a content page, describing what functions your program has, how to use the program (start, open, sort, search, print, quit, etc. etc.) You might even produce some on-line help that is in the program available (like help-balloons / fields / special pages / layout . . ) What is the quality of your offered help for the user?

You need to give the user some instructions about how he (or she) should work with your coursework program. How is it possible to enter data (what data?), what steps are to be taken, how can the user print out reports, etc. etc. You should show here e.g. some (hard)copies of your screen and explain what the user should do. On-line user help can easily be made in HyperCard and FileMaker. You can use here different cards, fields or Lay-Outs. Make it a separate section in your coursework, label it appropriate (like "User Manual")

0/1/2/3 points

## COURSEWORK EVALUATION

☚ Here you add your own **evaluation** for your solution to the problem.

You will need to describe how satisfied you are with your solution. Did you experience problems as you worked on your coursework, is your solution what you expected / aimed for? What did others think about your program (be as honest here, same as when you explain the teacher why you were late . .) What bugs (mistakes) are still there? Did you use the right program(s) or applications after all to solve your problem? Could you have chosen something else maybe better?

0/1/2/3 points

☚ Now it is time to list **options for improvement** . Here you can state what you have still in mind, what items, facilities etc. could be added!

What could be better in your solution. What possible future developments would be useful or even necessary? Will you use your coursework later in reality? Think about improvements in the software. You can mention also hardware that could be used in addition. Don't be shy here, go for the 2 points available!

0/1/2 points

## DEDUCTIONS FOR HELP

☞ This is the place where we can subtract points for any extra HELP you have received in the realisation of your coursework. Normally we won't subtract here anything. Help from books, tutorials, demo-programs etc. are all fine. Nothing against help, as long as you understand what you are doing and you are not just copying from others! But don't let daddy do the job. Instead have a look at old courseworks, talk to friends, and (!) don't wait till the last minute.

It is allowed for a deduction of up to **10** of your total points.

**DEDUCTIONS FOR TRIVIAL PROJECTS**

☞ This can be the case when you enter just a very simple and trivial project (anyone could do it!) A coursework for IGCSE must show that you really learned about modern Information Technology and you have good skills in different applications and software. Check out the syllabus to see what they say about it.

Here it would be allowed for a deduction of up to **25%** of your total points.

**IMPORTANT**

Total points for your coursework can be max. 50 points.
(counts for **25%** of your IGCSE-Grade, **75%** is for the theory paper of 21/2 hour).

Don't forget: We also have to testify that the coursework is your work only, and - not even partly - copied or done by somebody else!

Don't forget that you should make a good and clear documentation in total! You can achieve a better grade with a poor, but good documentated program, as with a smart and sophisticated program with weak and superficial documentation . . . . !

Frequently (all the time!!) use screen dumps (hard copies). They are often the only way how you can show what your program is doing. (Don't feel bad that you are killing trees when you print it all out. We will get the courseworks back from Cambridge and then you can recycle . . )

Don't forget to have a look at Chapter 27 in your computer book. It explains a lot about making a coursework.

If you feel uncertain, have a look at (e.g.) the chapters 9 (Databases), 13 (How to describe a computer system / program), 29 (Spreadsheet). (Stephen Doyle, Computer Systems for You, Stanley Thornes, ISBN 0-7487-2809-0)

Always meet the announced deadlines. Remember, the exam seems to be still far away, but remember the number of practical lessons! (see above). One more time: examiners in Cambridge won’t look at your database or website on-line. We cannot send floppies or ask them to look at a website. It’s all paperwork we send them . . . . So be prepared and work continuously on your documentation. Save it often, don’t take any risk.

Last but not least: 20+ students in a lab is for a teacher (any teacher!) not an easy workable situation. You will have to take a high responsibility for your own work! Don’t expect you can finish it all in the last week, it won’t work. Don’t waste time surfing on the Internet, use it for dedicated searches only. Don’t count pages, you must go for quality, not for quantity. Remember, all you never printed, was never sent in the first place . .

Good luck, happy working